IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
APPLICATION FOR UNITED STATES LETTERS PATENT

INVENTORS:

DAVID S. LAMPERT
RICHARD A. ASHBY
ROBERT FERNEKES
JAMES MEEK
JOHN JAUGILIS

TITLE:

PARCELIZED GEOGRAPHIC DATA
MEDIUM WITH INTERNAL SPATIAL
INDICES AND METHOD AND SYSTEM
FOR USE AND FORMATION THEREOF

ATTORNEYS:

Frank J. Kozak
BRINKS HOFER GILSON & LIONE
P.O. Box 10395
Chicago, Illinois 60610
(312) 321-4200

PARCELIZED GEOGRAPHIC DATA MEDIUM WITH INTERNAL SPATIAL INDICES AND METHOD AND SYSTEM FOR USE AND FORMATION THEREOF

BACKGROUND OF THE INVENTION

The present invention relates to system and method for facilitating access to and use of geographic data used with a navigation application program that provides navigating features and functions to an end user, and more particularly, the present invention relates to a geographic data medium wherein the data are organized into parcels and a method and system for the organization, storage, retrieval and use of the geographic data that facilitates the use thereof for various navigating functions provided by a navigation application program.

Computer-based navigation application programs are available that provide users with various navigating functions and features. For example, some navigation application programs are able to determine an optimum route to travel by roads between locations. Using input from a user, and optionally from equipment that can determine one's physical location (such as a GPS system), a navigation application program can examine various routes between two locations to determine an optimum route to travel from a starting location to a destination location in a geographic region. The navigation application program may then provide the user with information about the optimum route in the form of instructions that identify the maneuvers required to be taken by the user to travel from the starting location to the destination location. If the navigation system is located in an automobile, the instructions may take the form of audio instructions that are provided along the way as the user is traveling the route. Some navigation application programs are able to show detailed maps on computer displays outlining routes to destinations, the types of maneuvers to be taken at various locations along the routes, locations of certain types of features, and so on.

In order to provide these and other navigating functions, the navigation application program requires one or more detailed databases that include data which represent physical features in a geographic region. The detailed database may include data representing the roads and intersections in a geographic region and also may include information about the roads and intersections in a geographic region, such as turn restrictions at intersections, speed limits along the roads, the locations of stop

1   signs, street names of the various roads, address ranges along the various roads, and

2   so on.

3   One difficulty in providing geographic data for use by a navigation application

4   program relates to the efficient utilization of the available computer resources of the

5   navigation system on which the navigation application program is run. Computer-

6   based navigation application programs are provided on various platforms including

7   some with relatively limited computer hardware resources. For example, navigation

8   systems may be located in vehicles or may be hand-held. These types of navigation

9   systems typically have relatively limited computer resources, such as limited memory

10   and relatively slow I/O. In order to provide a high a level of functionality in such

11   systems, it is required that the available computer resources be used efficiently.

12   Given the relatively large size of the geographic database necessary to provide

13   a desired level of navigating functionality to the user, it is accepted that all the data

14   records for an entire geographic region cannot be loaded into the memory of the

15   navigation system at the same time. This is especially true for navigation system

16   platforms with limited resources, such as systems installed in vehicles or hand-held

17   systems. Due to the limited memory resources of these navigation systems, it is

18   necessary to load geographic data as needed from a storage medium, such as a

19   CD-ROM, into the memory of the navigation system for use by the navigation

20   application program. Unfortunately, as mentioned above, in these types of systems,

21   I/O access from a storage medium is also likely to be relatively slow. Thus, the

22   relatively limited memory resources combined with the relatively slow I/O can limit

23   the performance of some types of navigation systems, resulting in slow response.

24   Aside from being undesirable, slow response in a navigation system may render the

25   system useless for its intended purpose in certain circumstances. For example, if the

26   navigation system is installed in a vehicle, the driver may require information from

27   the navigation system about a desired route in a matter of seconds in order to utilize

28   the information while driving. If the navigation system requires more than several

29   seconds to calculate a route, the driver may have moved beyond the point at which the

30   routing information provided by the navigation system is relevant. Therefore, it is

31   important that navigation systems operate efficiently in order to provide navigating

32   information relatively quickly.

1   Navigation application programs may also be run on computer platforms that

2 have in general greater memory resources and faster I/O, such as personal computers

3. or on networks. Although these systems may have more and faster resources, the

4 considerations related to the efficient use of geographic data still apply, but on a larger

5 scale. With these types of systems, even greater functionality can be provided if the

6 limitations imposed by memory size and I/O are minimized.

7   Techniques have been devised or implemented to improve navigation system

8 performance by organizing, structuring, or arranging the geographic database or the

9 data in the database in particular ways. Because a navigation system uses geographic

10 data in certain known and expected ways to perform known functions, the geographic

11 data can be organized, structured, or arranged in a manner that facilitates their use in

12 these known ways by the navigation system. Included among the techniques that can

13 be used to facilitate the use of geographic data by navigation systems is parcelization.

14 When data are parcelized, the plurality of data records that together comprise the

15 geographic data are grouped together into separate groups or parcels. Another way

16 that geographic data can be organized to facilitate their use is to organize the data

17 spatially. When geographic data are organized spatially, features that are close

18 together physically in the geographic region are represented by data records that are

19 physically (or logically) close together in the database. Geographic data can be both

20 parcelized and spatial organized to take advantage both of these techniques.

21   Although techniques such as parcelization and spatial organization can

22 enhance the use of geographic data by a navigation system, use of these techniques

23 may be accompanied by some drawbacks. For example, spatial searches for

24 geographic data that do not coincide with parcel boundaries may require examination

25 of unnecessary data records. Accordingly, there continues to be a need for

26 improvement in the making and using of geographic databases in order to provide

27 better performance in navigation systems.

28   Accordingly, it is an objective to provide improvements in the storage or use

29 of geographic data that improves performance in a navigation system.

30

SUMMARY OF THE INVENTION

To address the above concerns, the present invention provides a navigable map database, stored on a computer-readable medium and used with a navigation application program, wherein the navigable map database includes data which are spatially parcelized into a plurality of parcels. Associated with each of the plurality of parcels is a first index that associates the area represented by the data in the parcel with a plurality of sub-areas formed of the area. Also associated with each of the parcels is a second index that associates each of the data in the parcel with at least one of the sub-areas.

According to a further aspect of the invention, there is provided a method for producing a navigable map database which is parcelized into a plurality of parcels, wherein each of the plurality of parcels includes a first index that associates the area represented by the data in the parcel with a plurality of sub-areas formed of the area and a second index that associates each of the data in the parcel with at least one of the sub-areas.

According to another aspect of the invention, there is provided a program and method for finding data in one or more parcels that matches a spatial search criterion using a navigable map database that is parcelized into a plurality of parcels, wherein each of the plurality of parcels includes a first index that associates the area represented by the data in the parcel with a plurality of sub-areas formed of the area and a second index that associates each of the data in the parcel with at least one of the sub-areas. The program and method use the first and second indices to identify which of the data in at least one of the plurality of parcels satisfy the spatial search criterion.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 illustrates a map showing a geographic region.

Fig. 2 shows an expanded view of a portion of the map of Fig. 1.

Fig. 3 is an illustration of a single road segment shown in the map of Fig. 2.

Fig. 4 is a diagram illustrating a geographic database for the geographic region illustrated in Fig. 1 and having separate subsets of data for use with navigation application programs.

1   Fig. 5 is a diagram similar to Fig. 4 illustrating both separate subsets of data

2 types and separate layers of data in some of the types.

3   Fig. 6 shows the map of a geographic region illustrating application of a

4 parcelization method to geographic data.

5   Fig. 7 shows a map of a portion of a geographic area with divisions into sub-

6 rectangles illustrating application of an embodiment that organizes geographic data

7 within a parcel.

8   Fig. 8 graphically illustrates a kd-tree structure used to represent the separation

9 of geographic data within a parcel using the divisions shown in Fig. 7.

10   Fig. 9 illustrates the kd-tree structure of Fig. 8 stored as data in the map

11 database.

12   Fig. 10 is a representation of an index structure used in connection with the

13 organization illustrated in Figs. 7-9.

14   Fig. 11 shows the map of Fig. 7 illustrating use of an embodiment for spatial

15 searches.

16   Fig. 12 is a block diagram showing a navigation system that includes an

17 embodiment of a programming application that accesses data arranged in accordance

18 with a parcelized map database having parcel internal data structures that facilitate

19 spatial searches.

20   Fig. 13 shows a map of a portion of a geographic area illustrating use of an

21 embodiment for spatial searches with more than one parcel.

22

23 DETAILED DESCRIPTION OF THE
24 PRESENTLY PREFERRED EMBODIMENTS

25 I.   **Overview of navigable map database**

26   In one present embodiment, the speed and/or functionality of a navigation

27 system can be enhanced by a combination that includes improvements in the storage,

28 arrangement, and/or structuring of the geographic data used by the system to facilitate

29 the use of the data by some of the functions in the navigation application program in

30 the systems that use the data. Based upon the manner in which the geographic data

31 are stored, arranged, and/or structured, functions in the navigation application

32 program that access the data can implement routines that exploit the improvements

1 incorporated into the geographic data. This combination can result in overall

2 improved performance by the navigation system.

3   Fig. 1 illustrates a map 10 showing a geographic region 12 and Fig. 2 shows

4 an expanded view of a portion 16 of the map 10. The portion 16 in Fig. 2 illustrates

5 part of the road network 20 in the geographic region 12. The road network 20

6 includes, among other things, roads and intersections located in the geographic region

7 12. As shown in Fig. 2 in the illustrated portion 16 of the map 10, each road in the

8 geographic region 12 is composed of one or more segments, 22(1), 22(2)... 22(n). In

9 one embodiment, a road segment represents a portion of the road. In Fig. 2, each road

10 segment 22 is shown to have associated with it two nodes 23: one node represents the

11 point at one end of the road segment and the other node represents the point at the

12 other end of the road segment. A single road segment 20 and its two associated nodes

13 23(A) and 23(B) are illustrated in Fig. 3. The node at either end of a road segment

14 may correspond to a location at which the road meets another road, e.g. an

15 intersection, or where the road dead ends. (An intersection may not necessarily be a

16 place at which a turn from one road to another is permitted, but represents a location

17 at which one road and another road have the same latitude and longitude.)

18   In addition, if the road segment 22 is other than straight (e.g., it bends, turns,

19 etc.), the road segment 22 may include one or more shape points 24 between its end

20 points 23. Shape points 24 provide geographic positions (i.e., latitudes, longitudes)

21 along the length of the road segment to accurately represent the true physical locations

22 of the road segment along its length. Shape points 24 are used to assist in vehicle

23 positioning, map display, etc.

24   In one type of geographic database, there is at least one database entry (also

25 referred to as "entity" or "record") for each road segment in a geographic region. This

26 road segment data record may have associated with it information (such as

27 "attributes", "fields", etc.) that allows identification of the nodes associated with the

28 road segment and/or the geographic positions (e.g. the latitude and longitude

29 coordinates) of the two nodes. In addition, the database road segment record may

30 have associated with it information (e.g. more "attributes", "fields", etc.), that specify

31 the speed of travel on the portion of the roadway represented by the road segment

32 record, the direction of travel permitted on the road portion represented by the road

1 segment record, the name of the road represented by the road segment record, what if

2 any turn restrictions exist at each of the nodes which correspond to intersections at the

3 ends of the road portion represented by the road segment record, street address ranges

4 of the roadway represented by the road segment record, and so on.

5   As described above, each segment data entity that represents an other-than-

6 straight road segment may include one or more shape points 24. The shape points 24

7 are represented by shape point data attributes that are associated with their respective

8 data segment entity. In collecting geographic data for use in navigation systems,

9 shape points are determined for segments of roads that bend or curve so that the

10 position of points along the road segment can be accurately determined.

11   Several navigation application functions require identification of a road

12 segment data record based upon the physical location of the portion of the road in the

13 geographic region which is represented by the road segment data record. Therefore,

14 the geographical position of the portion of the road represented by the road segment

15 data record is associated with the database entity for the road segment data record. In

16 one embodiment, the location of the road segment is identified by the positions of its

17 nodes. By convention or design, one of the two nodes associated with the road

18 segment may be used to identify the location of the road segment, although both

19 nodes or some other arrangement may be employed. The road segment data record

20 may have associated with it attribute information that allows identification of its

21 nodes.

22   In a geographic database that represents the region 12, there may also be a

23 database entry (entity or record) for each node in the geographic region. The node

24 data record may have associated with it information (such as "attributes", "fields",

25 etc.) that allows identification of the road segment(s) that connect to it and/or its

26 geographic position (e.g. the latitude and longitude coordinate).

27   Referring again to the map of Fig. 1, a plurality of locations 14 are shown to

28 be located in the geographic region 12. Each of the locations 14 represents a place or

29 point in the geographic area 12 at which there is located a feature about which it is

30 desired to include information in a geographic database. Each of these locations 14

31 has a unique physical location (latitude, longitude, and optionally absolute or relative

32 altitude) and each of the locations 14 can be uniquely identified by its two

1 dimensional (or three dimensional) geographic coordinates, (i.e., latitude, longitude,

2 and optionally altitude). A location 14 may correspond to one of the nodes located at

3 the end of road segment data entity, or may correspond to a point-of-interest, such as a

4 hotel or civic center, or may correspond to a point along a road segment at which the

5 direction of the road changes. The locations 14 may represent anything physically

6 located in the geographic area 12.

7

8 **II.   Separate subsets of geographic data**

9   One way that the accessing of geographic data can be enhanced for performing

10 various navigation functions is to provide separate collections or subsets of the

11 geographic data for use by each of the separate functions in the navigation application

12 program. Each of these separate subsets is tailored specifically for use by one of the

13 functions. For instance, the route calculation function normally uses only a portion of

14 all the information in the geographic database that is associated with a segment of a

15 road. For example, when the route calculation function is being run, it may require

16 information such as the speed along a road segment, turn restrictions from one road

17 segment to another, and so on. However, the route calculation function does not

18 normally require the name of the road to calculate an optimum route. Similarly, when

19 using the map display function, some of the information associated with a road

20 segment, such as the speed limits or turn restrictions, is not required. Instead, when

21 the map display function is run, it uses only a portion of the information associated

22 with the road segment, such as the shapes and locations of roads, and possibly the

23 names of the roads. Even further, when the maneuver function is being run, some of

24 the information associated with a segment of a road, such as the speed and turn

25 restrictions, is not required. Instead, when the maneuver function is being run, it uses

26 information that includes the name of the road represented by the road segment, the

27 address range along the road segment, any signs along the road segment, and so on.

28 Although there may be some overlap as to the types of information used by the

29 various navigation functions, some of the data used by any one of these navigation

30 functions is not used by another of the functions. If all the information relating to

31 each road segment were associated with it as a single data entry in a single database,

32 each data entity record would be relatively large. Thus, whenever any one of the

1   navigation functions accessed an entity record, it would have to read into memory a

2   significant amount of information much of which would not be needed by the

3   navigation function. Moreover, when reading the data entity from disk, relatively few

4   data entities could be read at a time since each data entity would be relatively large.

5   In order to provide the information in the geographic database in a format

6   more efficient for use by each of the navigation functions, separate subsets of the

7   entire geographic database for a given geographic region are provided for each of the

8   different types of navigation functions to be provided in the navigation application

9   program. Fig. 4 illustrates a geographic database 5 comprised of separate routing data

10   6, cartographic data 7 (for map display), maneuver data 8, and points-of-interest data

11   9. A geographic database may be defined with fewer or more subsets than these, and

12   other types of data may be defined and included.

13   Each subset of data includes only the data required to be used by a particular

14   navigation function. There is some overlap of data between each of these subsets,

15   with the result that some parts of the information may be included in more than one

16   subset. For example, both the road segment data entity in the routing data subset as

17   well as the road segment data entity in the cartographic data subset may include

18   attributes identifying the nodes located at the ends of the segments. Although this

19   duplication may result in an larger overall data storage requirement, each of the

20   navigation programs benefits from the resultant efficiency of handling smaller

21   amounts of data.

22   Providing for separate subsets of geographic data for each of the navigation

23   programs also takes into account that usage of each of these navigation functions

24   relates to the others of the navigating functions in expected ways. For example, a user

25   will often first want to view a present position, then enter a destination, then receive

26   instructions how to start toward the destination, then observe a map showing the

27   initial portion of the route, then receive further instructions, then have a map

28   displayed of the next portion of the route, and so on. Because of these type of

29   expected usages, dividing the data into subsets provides for efficient use of the data

30   when using each separate function.

31   Although the division of the geographic data into subsets provides for efficient

32   use of the data by each of the different navigation functions, it becomes necessary to

1   provide that the different navigating functions that use these different subsets of the

2   database work together. For example, in the example mentioned above, after a user

3   obtains a calculated route, it may be desired to display a map on a computer display

4   with the calculated route highlighted. In order to accomplish this, the routing subset

5   of geographic data is accessed first to obtain the routing road segment data entities for

6   the optimum route, and then the cartographic subset of the geographic database is

7   accessed to obtain the cartographic road segment data entities corresponding to the

8   routing data entities. To permit these data subsets to work together, index files 11

9   may be included that provide cross references, search trees, or other data finding

10   techniques.

11

12   **III.**   **Layering of geographic data**

13   Another way that the geographic data can be organized to enhance their use is

14   to provide the data in layers. Some of the navigation functions use the data at

15   different levels of detail. The map display function is an example of this type of

16   function. When using the map display function, it is sometimes desired to provide for

17   panning and zooming. Zooming can be done more efficiently if the data are organized

18   into layers, with greater detail at the lower layers and less detail at the higher layers.

19   When using the route calculation function, it is also advantageous to use the data at

20   different levels of detail. For example, when calculating a route between two

21   locations, it would be inefficient to examine all the possible road segments that

22   diverge from each intersection along the route, including secondary streets and alleys.

23   Instead, once a route is "on" a main road or expressway, it is generally preferable to

24   stay on main roads or expressways until it is necessary to exit to secondary roads as

25   the destination is approached. If the routing data are layered, higher layers that omit

26   secondary roads can be used when possible to minimize the possible road segments to

27   be investigated when calculating the route. Therefore, within some of the subsets of

28   data types, the geographic data are provided in separate collections or groups

29   corresponding to separate layers.

30   To implement layering, data entities, such as road segment data entities, are

31   provided with a "rank." The rank of a road segment may be related to its functional

32   class with road segments having a rank of "0" being slowest and narrowest, road

segments having a rank of "1" being larger and faster, road segments having a rank of "2" being major roads, and so on. The "rank" of a segment data entity also specifies the highest data layer in which a road segment entity exists. For example, referring to Fig. 5, the route calculation subset type of geographic data 6 may include five separate collections of the data, R0, R1, R2, R3, and R4, each with a different level of detail, which can be used by the route calculation function. Similarly, the cartographic subset type of geographic data 6 may include five separate collections of the data, C0, C1, C2, C3, and C4, each with a different level of detail, which can be used by the map display function.

In the routing type of the geographic database, layer 0 (R0) includes segment data entities corresponding to all the portions of all the roads in the geographic region. Level 1 of the routing data comprises a separate subset (or collection) of the routing data and includes only the routing segment data entities (and some or all of their corresponding routing data attributes) having a rank of level 1 or higher. Level 2 of the routing data comprises a separate subset of the routing data and includes only the routing segment data entities (and some or all of their corresponding navigation data attributes) having a rank of level 2 or higher, and so on.

Similarly, the cartographic subset of geographic data may include separate collections (layers) of the data used by the map display function, each with a different level of detail. In the cartographic subset of the geographic data base, layer 0 includes segment cartographic data entities (and corresponding data attributes) corresponding to all the portions of all the roads in the geographic region. Level 1 of the cartographic data comprises a separate subset of the cartographic data and includes only the cartographic segment data entities (and corresponding data attributes) having a rank of level 1 or higher, and so on. Using these different layers of cartographic data, the map display function can provide rapid panning and zooming.

Although the organization of some of the data into layers results in some duplication of the data, the increased efficiency provided by layering generally offsets any disadvantages. As with the use of separate types of data mentioned above, the need arises to allow these layers to work together. Index files 11 which include cross references, search trees, or other finding techniques, may be provided for this purpose.

IV.   Spatial access to geographic data

Organizing the data into subsets or types and layering the data of some of the types provide separate collections of the data in sizes that are more manageable by each of the navigation functions. With respect to some subset types and layers of these types, the data can be further organized to facilitate spatial access.

Several of the navigation functions provided in a navigation system may require access to the geographic data spatially. One way this arises is that a function in a navigation application program requires finding a data entity record in a geographic database given the physical location represented by the data entity in the geographic region. The data entity may be a road segment record that represents a portion of a road in the geographic region and the function may require finding the road segment record based upon the physical location in the geographic region of the portion of the road represented by the road segment record. Another way spatial access arises is when a function in a navigation application program requires finding several or all of a type of data records located close to a location in the geographic region or within a defined area in the geographic region. For example, a function may require all road segment records encompassed within a rectangle defined by geographical coordinates $(x, x+n)$ latitude and $(y, y+m)$ longitude.

The route calculation function is a function of the type that may access geographic data spatially. This function provides a user of the system with an optimum route for traveling from one location in a geographic area to a destination location. In order to determine the optimal route, the route calculation function requires access to road segment data entities to determine certain data attributes, such as speed limits, turn restrictions, and so on, associated with the road segment data entities along the various possible routes between the starting location and the destination location. For example, as part of the route calculation procedure, when the route calculation function determines the optimum road to take from an intersection, it may access all the roads that lead from the intersection. Thus, all the road segment records that represent portions of roadways that meet at the intersection are accessed and examined. The route calculation function requires that the road segment records be accessed by locations in the geographic area of the roadways to which they

1 correspond. The map display function and the maneuver generation function are also

2 functions that may require access to geographic data spatially.

3 Assuming that all the data records for a given entire geographic region cannot

4 be loaded into memory at the same time due to limited memory resources of the

5 navigation system in which the navigation application program is being run, it would

6 be desirable to load into memory only those data that are needed. Since some of the

7 navigation functions require accessing data spatially, it would be advantageous to

8 provide a means to load data into memory based generally upon the physical

9 geographic locations of the features which the data represent or upon the geographical

10 proximity of the features which the data represent. This can be done by parcelizing

11 the data so that they are organized in the database and/or on the medium based upon

12 the geographic locations of the features which are represented by the data. Different

13 method for parcelizing geographic data are described below.

14

15 **V.   Parcelization of geographic data**

16 There are several factors that can affect the operation and performance of a

17 navigation application program when using geographic data. In order to provide a

18 reasonably high level of functionality, a relatively large database may be provided.

19 Storage media, such as CD-ROM or PCMCIA cards, are capable of handling

20 databases of the size and complexity sufficient to provide suitable functionality.

21 However, accessing these types of media can be relatively slow. Because navigation

22 systems may be installed in vehicles, or may be hand-held, the hardware resources of

23 the navigation system may be limited. Navigation systems installed in vehicles or

24 hand held units may have limited memory resources and relatively slow media access

25 rates.

26 As mentioned above, it is sometimes desirable to store the data representing

27 the roads and intersections based upon the physical proximity of the physical features

28 that they represent. In order to accomplish this, such data are organized into parcels

29 with each parcel of data including data which represent features which are located

30 physically proximate to each other in the geographic region. A parcel of data is

31 established to be the smallest quantity of data that can be accessed at a time. This

32 may relate to the quantity of data that can be accessed in a single disk access, although

1   it may be related to some other factor. For some types of media such as a CD-ROM, a

2   parcel may be established to a 16 Kilobyte quantity of data. (Other sizes of data may

3   be used including 1 K, 2 K, 4 K, 8 K, 32 K, and so on.)

4   (For purposes of forming the data into parcels, the data are first separately

5   organized into the different types, as described above, based upon the functions that

6   access them, such as routing, map display, and maneuver generation. Further, the data

7   are also organized into layers, as mentioned above, based upon rank. Therefore, this

8   description of parcelization refers to the level 0 routing data although it is applicable

9   to other types and levels of data as well.)

10   There are a number of different techniques for parcelizing geographic data.

11   One parcelization procedure is disclosed in the copending application Ser. No.

12   08/740,295, filed October 25, 1996, the entire disclosure of which is incorporated by

13   reference herein. The parcelization method disclosed in Ser. No. 08/740,295 is briefly

14   described below.

15

16   *A.*   *First parcelization process.*

17   A difficulty to be addressed when parcelizing geographic data is that some of

18   the parcels may not have a desired fill percentage. This may occur in part because a

19   geographic region is usually not uniformly-featured. Some parts of a geographic

20   region have more features than other parts and therefore there is more data in the

21   database representing those denser parts of the region than the parts that are less

22   densely-featured. Then, a parcelization procedure that forms parcels of data based

23   upon grouping together data which represent features encompassed within separate

24   rectangular areas formed by a simple bisection of the entire geographic region may

25   result in some rectangles containing significantly less data than the maximum parcel

26   size.

27   The parcelization method described in Ser. No. 08/740,295 overcomes this

28   disadvantage by using both a normal bisecting procedure and a special dividing

29   procedure to parcelize geographic data. According to Ser. No. 08/740,295, if the data

30   representing the features encompassed in a rectangle are greater than a predetermined

31   multiple of the maximum parcel size, the rectangle is bisected into two equal sized

32   sub-rectangles. However, if the data representing the features encompassed in a

1   rectangle are less than the predetermined multiple of the maximum parcel size, the

2   rectangle is divided using a special division procedure that increases the likelihood

3   that the parcels formed will have a desired fill percentage.

4   When data representing the features encompassed in a rectangle are less than a

5   predetermined multiple of the maximum parcel size, trial divisions of the rectangle are

6   examined at locations in addition to a division by bisection. In one embodiment,

7   divisions are examined at ½ along the rectangle (i.e. a bisection), ¼ along the

8   rectangle, ¾ along the rectangle, 1/8 along the rectangle, 3/8 along the rectangle, and

9   so on, through 31/32 along the rectangle. A division is selected so that further

10   divisions will result in the minimum number of parcels being formed of the data.

11   The parcelization procedure disclosed in Ser. No. 08/740,295 may start with a

12   rectangle that encompasses the entire geographic region, or preferably, it starts with a

13   plurality of starting rectangles defined by a regular grid which is overlaid on the

14   region. Such a grid is formed so that a starting rectangle is the largest rectangle

15   allowed such that the data representing the features encompassed therein are permitted

16   to form a parcel. Using a regular grid to define starting rectangles facilitates

17   parcelization since it is unlikely that any of the first several divisions of a rectangle

18   encompassing the entire region will form rectangles small enough to form a parcel.

19   Thus, overlaying a grid represents several bisections of the entire region.

20   This parcelization procedure uses a minimum enclosing dividable-tile ("di-

21   tile") for purposes of determining the point at which a division of any rectangle (or

22   sub-rectangle) is made. Referring to Fig. 6, in this parcelization method, a minimum

23   enclosing di-tile 200 is determined that encompasses a minimum bounding rectangle

24   202. A di-tile refers to an area of dimensions $2^I \times 2^J$ that includes all map data between

25   latitudes $M \times 2^I$ navigation units and $(M+1) \times 2^I$ navigation units and between

26   longitudes $N \times 2^J$ navigation units and $(N+1) \times 2^J$ navigation units, where M and N

27   are integers, and I and J are positive integers). The navigation units 1, 2, ... an so on,

28   may represent units equal to 1/100,000th of a degree. (To allow di-tiles to overlap

29   "0", i.e. latitude or longitude equal to "0", di-tiles may also have dimensions between

30   -2I, +2I, where I ≥ 17.)

31   One way of determining a minimum enclosing di-tile is to define acceptable

32   intervals and to require that the minimum enclosing di-tile have as its sides only

1 acceptable intervals. Acceptable intervals are defined in both directions of latitude

2 and longitude. (Any arbitrary starting location may be chosen, but in a preferred

3 embodiment, acceptable intervals conform to conventional latitude and longitude

4 starting locations, i.e. the equator and Greenwich.) Acceptable intervals may be

5 defined to include only powers of 2, for example: 0-1, 2-3, 4-5, 6-7, ... , 0-3, 4-7,

6 8-11, 12-15, ... , 0-7, 8-15, 16-23, 24-31, ... , 0-15, 16-31, 32-47, 48-63, ... , and so on

7 (in navigation units). Acceptable intervals include $\{M*2I, (M+1)*2I\}$ where $M \in Z$

8 (where M is a member of the set of all integers Z), $I \geq 0$; or acceptable intervals may

9 also include $\{-2I, +2I\}$ where $I \geq 17$ in order to obtain intervals that overlap latitude or

10 longitude equal to 0.

11 　　　　In this embodiment, the sides of the minimum enclosing di-tile for the

12 minimum enclosing rectangle are required to be acceptable intervals. Therefore, in

13 this embodiment, the east-west coordinates of the initial di-tile are multiples of 2I

14 units, and the north-south coordinates of the initial di-tile are multiples of 2J units. (I

15 and J are integers so that the east-west length of the initial di-tile may have a

16 dimension in units of 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, or 1024, and so on, and the

17 north-south length of the initial di-tile may have a dimension in units of 1, 2, 4, 8, 16,

18 32, 64, 128, 256, 512, or 1024 and so on, for example).

19 　　　　Once the minimum enclosing di-tile is established, the data can be parcelized.

20 In one alternative, the parcelization process can begin by applying the regular division

21 procedure, described below, to the minimum enclosing di-tile. Alternatively, the data

22 in the coverage area are first examined based upon an organization of the data into a

23 regular grid of rectangles formed from the minimum enclosing di-tile. This is

24 equivalent to bisecting the minimum enclosing di-tile and then the rectangles (or

25 squares) formed therefrom a number of times until a regular grid of rectangles results.

26 Each of the rectangles in this grid may be referred to as an "initial tile" (corresponding

27 to the "starting rectangle", above) The initial tile size is determined to be the largest

28 geographic area allowed to be represented by one parcel at any layer of any of the

29 types of data in the geographic database. In one embodiment, one fixed initial tile

30 size is defined for all regions throughout the country so that regions can be more

31 easily merged. In one present preferred embodiment, each of the initial tiles is of a

32 fixed, predetermined size of 2^{17} navigation units by 2^{17} navigation units.

1   The placement of the boundaries of the grid is determined in order to include

2   the region of the minimum bounding rectangle ("MBR") within a minimum enclosing

3   di-tile. The MBR is the rectangle formed by the north-south line through the

4   minimum longitude (corresponding to the easternmost node encompassed in the

5   geographic region), the east-west line through the minimum latitude (corresponding to

6   southern-most node encompassed in the geographic region), the east-west line through

7   the maximum latitude (corresponding to northernmost node encompassed in the

8   geographic region), and the north-south line through the maximum longitude

9   (corresponding to the westernmost node encompassed in the geographic region). The

10 grid boundaries are defined so as to correspond to the minimum enclosing di-tile

11 when the grid is overlaid on the region. All the spatial data are encompassed and the

12 initial tiles have a size as described above. In a preferred embodiment, the placement

13 of the grid boundaries also conforms to the acceptable intervals, described above.

14   As mentioned above, a purpose of parcelizing the data is to include in each

15 parcel an amount of data that is close as possible to, but not in excess of, a

16 predetermined maximum parcel amount. For example, the predetermined maximum

17 amount may be 16 Kilobytes.

18   Each one of the initial tiles in the grid is examined as a "trial parcel" to see if

19 the amount of data in it fits into a single parcel. If the data within the "trial parcel",

20 including any parcel overhead (such as index information and headers), would

21 (accounting for data compression, if used) be less than or equal to the maximum

22 parcel amount, then a parcel is constructed with that initial tile and no division of that

23 initial tile for that particular data type is performed. On the other hand, any "trial

24 parcel" that includes an amount of data that exceeds the predetermined maximum

25 parcel amount is divided using one of the two following procedures as a function of

26 the amount by which the data in the "trial parcel" exceeds the desired maximum

27 amount. (In a preferred embodiment, an estimation technique, described below, is

28 used in determining trial parcels. The estimation technique takes into account parcel

29 overhead and compression without actually performing all the steps necessary to form

30 a parcel.)

31   *Regular dividing procedure.* If the amount of data in a trial parcel exceeds the

32 maximum parcel amount by a predetermined multiple, the trial parcel is divided into

1   two rectangles. In a preferred embodiment, the division of the trial parcel into two

2   rectangles is carried out by first determining the minimum enclosing di-tile for the

3   trial parcel (in the manner described above with the initial tile), bisecting the

4   enclosing di-tile, and then dividing the trial parcel where the line of bisection of the

5   di-tile intersects the trial parcel. Alternatively the trial parcel may itself simply be

6   bisected. (It is noted that a bisection of the enclosing di-tile will not always bisect the

7   trial parcel, but instead may divide the trial parcel at an off-center location. For ease

8   of reference herein, such division of the trial parcel will nonetheless be referred to as

9   "bisection"). In either event, the line of bisection of the di-tile will be in either the

10   longitudinal or latitudinal direction. In a one embodiment, the di-tile is bisected in

11   whichever of the longitudinal or latitudinal divisions minimizes the maximum aspect

12   ratio of the two resulting rectangles of the trial parcel.

13   Each of these resulting rectangles is then examined as a trial parcel, as

14   described above, and bisected if the data contained in it exceeds a predetermined

15   multiple of the maximum parcel amount. Each of these sub-rectangles is also then

16   examined as a "trial parcel", as described above, and the process continues until the

17   amount of data in a rectangle or sub-rectangle is less than the predetermined multiple

18   of the maximum parcel amount. The predetermined multiple is chosen based upon a

19   desired minimum fill percentage for each parcel. In one embodiment the desired

20   minimum fill percentage is 80% and the predetermined multiple is 3.2 which can be

21   derived from the following table (where D is an unacceptable data size and P is the

22   maximum parcel size):

Table of unacceptable sizes D
$0 < D < 0.8 \times P$
$P < D < 1.6 \times P$
$2 \times P < D < 2.4 \times P$
$3 \times P < D < 3.2 \times P$

23

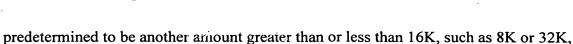24   The list stops at this point, because the next entry would be the empty range

25                          $4 \times P < D < 4 \times P.$

26   In one embodiment, the maximum parcel amount is predetermined to be

27   16 Kilobytes of data. (In alternative embodiments, the maximum amount may be

1   predetermined to be another amount greater than or less than 16K, such as 8K or 32K,

2   or even amounts greater or less than these). Thus, trial parcels are bisected according

3   to the regular procedure when their data content exceeds 51.2 kilobytes. When the

4   amount of data in any trial parcel is less than the predetermined multiple amount,

5   further subdivisions of the trial parcel follow the custom division procedure, described

6   below.

7   *Custom division procedure*. If the amount of data in any trial parcel exceeds

8   the maximum parcel amount, but is less than the predetermined multiple of the

9   maximum parcel amount (e.g., 16Kb < x < 51.2Kb), the following custom division

10 procedure is used: Further divisions of the trial parcel are not necessarily bisections,

11 but rather are made in a manner that tends to minimize the number of parcels created.

12 This has the effect of minimizing both the space needed to store the parcels and

13 wasted space within the parcels.

14 For example, given a trial parcel that contains data equal to 3.6 times the

15 maximum parcel size or amount, it should be possible to fit this data into four parcels.

16 However, bisection of the trial parcel may divide it into two rectangles of 1.2 times

17 the maximum parcel size and 2.4 times the maximum parcel size, respectively, which

18 would then end up in a minimum of five parcels if bisection were used to divide each

19 of the rectangles. Therefore, subdivisions of the rectangle at this stage are made with

20 the goal of minimizing the number of parcels created, but with the restriction that the

21 division line not be arbitrary. More particularly, where a trial parcel has a data

22 content that is greater than the maximum parcel size, but not in excess of the

23 predetermined multiple thereof, the trial parcel is divided at a division of 2^{-x} along one

24 of its dimensions. In a present preferred embodiment, X={1,2,3,4,5}. Thus, the trial

25 parcel is divided at 1/2 or 1/4 or 1/8 or 1/16 or 1/32 divisions of its width. For

26 example, a trial parcel may be divided into two rectangles with widths equal to 5/8

27 and 3/8, respectively, of the width of the trial parcel. This custom division may be

28 applied directly to the dimensions of the trial parcel, or, in a present preferred

29 embodiment, it may be applied to the dimensions of, a minimum enclosing di-tile of

30 the trial parcel. In the latter case, the trial parcel is divided where the division line of

31 the di-tile intersects the trial parcel. In either event, the division line will be in either

32 the longitudinal or latitudinal direction.

1   Candidate division lines are examined as follows: First, divisions are made at

2   each of the specified 2^{-x} divisions along both the longitudinal and latitudinal widths of

3   the trial parcel. For each such division, the aspect ratios (defined as the ratio of the

4   larger dimension to the smaller dimension) of each of the two resulting rectangles are

5   determined, and the greater of the two is identified. The greatest aspect ratios

6   identified for each of the candidate division lines are then compared, and the

7   candidate division lines are ordered from smallest to greatest of such aspect ratios.

8   The rectangles resulting from the candidate division lines are then examined,

9   beginning with the first candidate line in the ordered list. The candidate division line

10   chosen for dividing the trial parcel is the first one in the list encountered where the

11   data content in one of its two resulting rectangles is less than or equal to a multiple

12   (such as two times) of the maximum parcel size and greater than or equal to a

13   minimum fill percentage times such amount. For example, the division line is chosen

14   to include in one of the resultant rectangles an amount between 1.6 and 2.0 times the

15   maximum parcel amount. This should enable making one more division of the

16   rectangle to form two further sub-rectangles each with a fill percentage greater than

17   80% (.8) of the maximum parcel amount. Each of these resultant sub-rectangles with

18   a fill percentage between 80% and 100% of the maximum parcel amount is formed

19   into a parcel. If no candidate division line meets this criterion, then the first candidate

20   division line (i.e. the one with the smallest maximum aspect ratio) is used to divide

21   the given rectangle.

22   *Example*

23   The following describes how, during parcelization, a determination is made

24   when to stop bisecting trial parcels, and to start the custom procedure of evaluating

25   candidate divisions of 1/32, 1/16, 3/32, 1/8, 5/32, 3/16, 7/32, 1/4, 9/32, 5/16, 11/32,

26   3/8, 13/32, 7/16, 15/32, 1/2, 17/32, 9/16, 19/32, 5/8, 21/32, 11/16, 23/32, 3/4, 25/32,

27   13/16, 27/32, 7/8, 29/32, 15/16, and 31/32 along both the longitudinal and latitudinal

28   widths of the trial parcel.

29   A target parcel fill percentage F is chosen. For the sake of example, let F be

30   0.8 (80 percent). As mentioned above, a maximum parcel size P is also determined.

31   P is expressed in bytes and is the maximum amount of data that can be put into a

1   parcel. Optimally, therefore, it is desired to create parcels in the range of $F \times P$ bytes

2   and P bytes in size.

3   If a trial parcel having a data size D is in the range $P < D < 1.6 \times P$, then it is

4   not possible for parcels to be created therefrom whose data sizes fall in the target

5   range. If the trial parcel is divided such that one resulting rectangle has a data size

6   greater than or equal to $0.8 \times P$, then the other one has a data size less than $0.8 \times P$.

7   This process can be extended to give the list of non-acceptable data sizes in the

8   Table of Unacceptable Sizes, above. From the above list a complementary list of

9   acceptable data sizes can be generated:

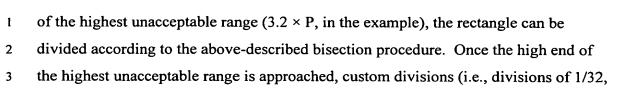| Table of Acceptable Data Sizes D: |
| --- |
| $0.8 \times P \leq D \leq P$ |
| $1.6 \times P \leq D \leq 2 \times P$ |
| $2.4 \times P \leq D \leq 3 \times P$ |
| $3.2 \times P \leq D$ |

10

11   The above list corresponds to a fill percentage F equal to 0.8. Other fill

12   percentages generate different lists of acceptable or unacceptable data sizes. In

13   general, the list of unacceptable data sizes is in the following form:

Unacceptable Data Sizes D:
$0 < D < F \times P$
$P < D < 2 \times F \times P$
. . .
$n \times P < D \leq (n+1) \times F \times P$
. . .

14

15   . . . continuing until an empty range is reached.

16

17   The above is used as follows in the custom procedure for forming the parcels:

18   The data sizes of each of the two rectangles resulting from a trial parcel division

19   should fall within one of the acceptable ranges (whenever possible). In practice, this

20   means that as long as the data size in a rectangle is somewhat larger than the high end

1 of the highest unacceptable range (3.2 × P, in the example), the rectangle can be

2 divided according to the above-described bisection procedure. Once the high end of

3 the highest unacceptable range is approached, custom divisions (i.e., divisions of 1/32,

4 1/16, 3/32, 1/8, etc. in this example) are considered.

5 Candidate division lines are examined and compared in the manner described

6 above, and the one selected for division is where the following criterion is met. The

7 amount of data falling into each of the two sub-rectangles of the trial parcel is of a size

8 such that it is theoretically capable of, in turn, being subdivided into rectangles that,

9 on the average, achieve a specified minimum parcel data fill percentage. Some cases

10 may occur in which the above criterion cannot be met.

11 The data are divided at the division line chosen in the custom division

12 procedure, and, for each of the two sub-rectangles created, the custom division

13 process is repeated as necessary. As mentioned above, the bisection and custom

14 division procedure can be applied either directly to the trial parcel or to the minimum

15 enclosing di-tile of the trial parcel, although the latter is preferred. It is noted that in

16 some cases the minimum enclosing di-tiles are exactly equal to trial parcel

17 boundaries. This may occur with respect to the initial tiles. The utility of defining the

18 divisions in terms of the minimum enclosing tiles is that a tile can be repeatedly

19 divided in half evenly, whereas a trial parcel rectangle of arbitrary dimensions cannot.

20 Another advantage is that this procedure facilitates processing at boundaries between

21 different databases. The custom division of a trial parcel at a 2^{-X} division of the tile's

22 side is equivalent to a sequence of from 1 to X bisections. Consequently, the division

23 lines, and therefore the resulting sub-rectangles, can be represented in a minimal

24 number of bits (5 bits for 1/32 divisions, as opposed to up to eight or sixteen bytes to

25 define an arbitrary rectangle).

26 (In examining amounts of data included in rectangles, a convention is

27 established that any data entity that is located exactly on a dividing line is included

28 with the data in the rectangle "to the right" ("east") of the line, if the dividing line is a

29 north-south line, and with the data "above" ("north of") the line, if the dividing line is

30 an east-west line.)

31 The above procedure is performed on all the initial tiles (and, where necessary,

32 all resulting rectangles) in the grid.

- 22 -

1

2   *B.   Second parcelization process*

3   Another parcelization procedure is described in the patent application entitled

4   "METHOD AND SYSTEM FOR PARCELIZATION OF GEOGRAPHIC DATA

5   FOR STORAGE AND USE IN A NAVIGATION APPLICATION" filed September

6   5, 1997, Attorney Docket No. 7117-61, the entire disclosure of which is incorporated

7   by reference herein. This parcelization method is a further improvement and can be

8   used in conjunction with the parcelization procedure described in Ser. No.

9   08/740,295.

10   In the parcelization method described in Ser. No. 08/740,295, it was proposed

11   as an example that parcels at least 80 percent filled are desired. For a parcel size "P",

12   a rectangle containing 2.7 x P bytes of data can be divided in different ways. For

13   example, it can be divided into two rectangles containing 1.35 x P bytes of data each.

14   Each sub-rectangle can then be divided into two sub-rectangles to obtain rectangles

15   encompassing data less than the optimal parcel size, resulting in a total of four parcels

16   averaging 68 percent filled. Alternatively, the initial division could be into sub-

17   rectangles with a 1.8 x P and 0.9 x P bytes of data, and then the former sub-rectangle

18   can be divided into two sub-rectangles of 0.9 x P bytes each, for a total of three

19   parcels 90 percent filled.

20   The decision made in the above example is based on a table of acceptable data

21   size ranges according to the following table:

Table 1	
Minimum	Maximum
0.8 x P	1.0 x P
1.6 x P	2.0 x P
2.4 x P	3.0 x P
3.2 x P	-------

22

23   In the parcelization method described in Ser. No. 08/740,295, the above table

24   is used to determine that a dividing line is acceptable based upon when it divides a

25   rectangle into two sub-rectangles each of whose data sizes falls within one of the

26   ranges above.

1   In many situations, the decision will also be constrained by the accuracy with

2   which a data size of a parcel can be estimated before the parcel is created. The

3   parcelization method disclosed in the aforementioned patent application entitled

4   Attorney Docket No. 7117-61 addresses this consideration as follows. If it is assumed

5   that the data size can be estimated to within an accuracy of 4 percent, then to provide

6   for parcels falling between 80 percent and 100 percent filled, without exceeding the

7   optimum size, it is required to modify the above table to take a 4 percent error into

8   account:

9

Table 2	
Minimum	Maximum
0.84 x P	0.96 x P
1.68 x P	1.92 x P
2.52 x P	2.88 x P
3.36 x P	3.84 x P
4.20 x P	4.80 x P
5.04 x P	5.76 x P
5.88 x P	--------

10

11   However, in some situations dividing lines can be considered that produce

12   sub-rectangles falling within acceptable ranges in the first table, but not the second

13   table. Therefore, a hierarchy of two or more tables is used. In the example here, if a

14   dividing line is found whose sub-rectangles fall within acceptable ranges in the second

15   table, that dividing line would be selected. Otherwise, a dividing line whose sub-

16   rectangles fall within acceptable ranges in the first table would be selected.

17   It could also happen that, for a particular dividing line, one sub-rectangle

18   might be acceptable based on both the first and second tables, and the other acceptable

19   based on the first table only. Each dividing line is therefore scored as follows: each

20   of the two sub-rectangles are scored according to the tables its data size falls into, and

21   the score of the dividing line is defined as the sum of the two sub-rectangle scores. In

22   this example, a dividing line both of whose resulting sub-rectangles fall into a range in

23   the second table would score highest, and a line both of whose sub-rectangles fall into

1   a range in the first table but not in the second table would score lowest. The best

2   dividing line is then selected using these scores.

3   Aspect ratio (the ratio between the sides of a rectangle) is also taken into

4   account, as follows: For a given dividing line, the aspect ratio of each of the two sub-

5   rectangles is determined, and the aspect ratio for the dividing line is defined to be the

6   least desirable (largest) of the two sub-rectangle aspect ratios. Only dividing lines

7   whose aspect ratios are smaller than a fixed maximum aspect ratio are considered

8   qualified candidates. Of the qualified candidates, the one with the best score will be

9   selected.

10   A further refinement of this method is as follows: In some situations, it may

11   not be possible to achieve the target fill percentage (e.g. 80-100 percent), because real-

12   world data are too discontinuous. When this occurs, some the data sizes of some

13   parcels will fall in the range between (0 - 80) percent of the optimal parcel size.

14   Under these circumstances, it would be preferable if the parcel size were equal to the

15   optimal percent size divided by 2K. If a buffer size in the memory of the navigation

16   system is P, then it is possible to make more efficient use of buffer space if the parcel

17   sizes are P, P/2, P/4, etc. (rather than ¾ x P, 5/8 x P, etc.). The reason for this is as

18   follows. When a record size is P/2K , it is more likely that a buffer slot can be found

19   for it without rearranging other records in the cache of the memory of the navigation

20   system. Therefore, when the data size of the rectangle being divided is less than 2 x

21   P, the tables described above are augmented with additional ranges representing the

22   optimal parcel size divided by 2K, for K in the range from 1 to about 3. In the above

23   example, the two tables would be augmented with the following ranges:

24

First Table:	
Minimum	Maximum
0.1 x P	0.125 x P
0.2 x P	0.25 x P
0.4 x P	0.5 x P

25

26

Second Table:	
Minimum	Maximum
0.105 x P	0.12 x P
0.21 x P	0.24 x P
0.42 x P	0.48 x P

1

2   In the First Table, above, acceptable ranges are defined for 80% to 100% of

3   1/8 times the maximum parcel size (0.1P-0.125P), 80% to 100% of 1/4 times the

4   maximum parcel size (0.2P-0.25P), and 80% to 100% of 1/2 times the maximum

5   parcel size (0.44P-0.5P). In the Second Table, above, acceptable ranges are defined

6   that take into account the approximately 4% tolerance for errors in estimating. In the

7   Second Table, above, acceptable ranges are defined for 84% to 96% of 1/8 times the

8   maximum parcel size (0.105P-0.12P), 84% to 96% of 1/4 times the maximum parcel

9   size (0.21P-0.24P), and 84% to 96% of 1/2 times the maximum parcel size (0.42P-

10   0.48P). Thus, the parcelization method described in the aforementioned patent

11   application , Attorney Docket No. 7117-61, facilitates use of the geographic data for

12   many types of navigation functions.

13

14   *C.*   *Subsequent parcelizations*

15   Each of the different types of data (routing, cartographic, maneuver, and so on)

16   and each layer of some of these types of data, are separately parcelized using any of

17   the procedures described above. The different types of data and the separate layers of

18   data of the each type may be parcelized in parallel. This means that after one type of

19   data is parcelized, parcelization of another type of data is performed using the same

20   rectangle boundaries (i.e. dividing lines) that were used for the first type. Parcelizing

21   may start with the type of data expected to be most dense. A subsequent parcelization

22   of a less dense type of data will follow the same rectangle boundaries that were used

23   for the first type of data. However, since the second type of data may be less dense, it

24   may be possible to form a parcel of the second type of data without making as many

25   divisions of rectangles encompassing the features represented by the data. This would

26   have the result that a parcel of one type of data may represent geographic features

27   encompassed by a larger sized rectangle than a parcel of another type of data.

1　However, whenever a division of data is to be made, it is made along the same

2　rectangular boundary division as was made in the prior type of data.

3　　　When parcelizing one type of data after a first type of data has already been

4　parcelized, it may sometimes occur that the subsequent type of data is more dense in a

5　portion of the region than the type of data which has already been parcelized. In this

6　situation, the smallest rectangle formed by the prior parcelization of data may not be

7　small enough for forming a parcel of data of the subsequent type of data. In such a

8　situation, the subsequent type of data may be further parcelized. A procedure for

9　performing such an additional parcelization is described in the aforementioned patent

10　application Attorney Docket No. 7117-61.

11　　　As mentioned above, some of the types of data may be layered. Each layer is

12　treated as a separate type of data for purposes of parcelization. Accordingly, for

13　higher layers of data, i.e. layers having fewer data entities in them, parcelization is

14　performed on the data, starting with a collection of all the data for the entire region

15　that meet the criteria for inclusion in that layer. When parcelizing higher layers of a

16　type of data, the same rectangle boundaries determined for the parcelization of layer 0

17　are used. Since higher layers of a type of data are less dense than layer 0 (which

18　includes all the data entities of a type for all ranks), parcels may be formed by

19　rectangles of a larger size.

20

21　　　D.　　*Alternative parcelization methods.*

22　　　In addition to the above described ways to parcelize the geographic data, there

23　are other ways in which the data can be parcelized. For example, a simple

24　parcelization method may provide for separating the geographic data into a plurality

25　of parcels wherein the data in each parcel represent features encompassed within a

26　separate one of a plurality of regular sized rectangles which together form a regular,

27　rectangular grid over the geographic region. Another method for parcelization is to

28　separate the data into parcels encompassed within rectangular areas where each of the

29　rectangles is formed by a bisection of the another rectangle until a parcel size below a

30　maximum threshold is obtained. Some of these parcelization procedures are described

31　in the copending patent application entitled "METHOD AND SYSTEM FOR

32　FORMING AND USING GEOGRAPHIC DATA" filed September 5, 1997,

1 Attorney Docket No. 7117-62, the entire disclosure of which is incorporated by

2 reference herein. Still another method of parcelization to which the disclosed subject

3 matter can be applied is described in U.S. Pat. No. 4,888,698.

4

5   *E.*   *Ordering of parcels*

6   As the parcels are formed for all the types of data and for all the layers of each

7 type, the parcels are ordered. Various types of ordering may be used. In general, it is

8 preferred that the parcels be ordered in a manner that minimizes searches for data. In

9 some of the functions in a navigation application program, there is sometimes a

10 requirement to access data that represents features along routes or paths across parts of

11 the geographic region. This may occur when calculating a route across the geographic

12 region or when panning across the region. Sometimes these routes or paths extend

13 over more than one of the rectangles from which the parcels of data were formed.

14 Accordingly, starting with data in one parcel, there is a requirement for accessing the

15 data in another parcel formed from a rectangle which is located adjacent to the

16 rectangle from which the first parcel was formed. Since each rectangle may have

17 several other rectangles adjacent to it, there is a need for ordering the parcels formed

18 by the rectangles to minimize searches.

19   One way to order parcels is to use a depth-first ordering from the kd-tree index

20 within each parcel type and layer. This provides an ordering similar to Peano-key

21 ordering. Parcels are stored on disk in this approximate Peano key order. An index in

22 the form of a kd-tree can be used to access parcels containing route calculation data

23 spatially. This index is useful for initial location of an arbitrary position, such as

24 when a route guidance program in a navigation system initially locates the map data

25 corresponding to a current vehicle position.

26   Once a position is known, the disk location of connected or nearby map data

27 can sometimes be found using data internal to a parcel. For example, if a data record

28 for the right node of a road segment in parcel A is outside of parcel A, the disk

29 location of the parcel containing that node can be found within parcel A.

30

1 **VI.   Organization of data within parcels**

2   Organization of geographic data into parcels using the processes described

3 above provides advantages that enable a navigation system that uses the geographic

4 data to find needed data more quickly thereby speeding up operation of the navigation

5 system. Organizing the geographic data within at least some of the parcels provides

6 further advantages that enable the navigation system using the data to operate even

7 more quickly. A present embodiment provides for an organization of the data within

8 at least some of the parcels and an index structure used with data organized in this

9 manner. Use of the present embodiment facilitates spatial searches for data within a

10 parcel, thereby enhancing operation of the navigation system that uses the data.

11   According to a present embodiment, in at least the bottom layer of one type of

12 data (for example, routing layer 0), and alternatively in other layers and in other types

13 of data, node data records are stored in Peano key order within the parcel based on

14 latitude and longitude. Segment data records are stored in approximate Peano key

15 order based on the latitude and longitude of the segment's left node. (A segment's

16 "left node" is the node whose coordinates are the further west of the two nodes

17 associated with the segment. If both nodes have the same longitude, then the node

18 whose coordinates are further south of the two nodes is selected. The use and

19 definition of the segment's left node for purposes of defining a segment's position for

20 storage purposes is arbitrary. For example, in alternative implementations, the right

21 node, the up node, or the down node may be used instead. However, whichever node

22 definition is uses, it should be applied consistently throughout the portion of the

23 database organized in this manner.)

24   Using this Peano ordering, all nodes and all segments in the region are given a

25 unique entity identifier. This is achieved as follows: (1) bottom layer route

26 calculation data are parcelized; (2) parcels are then processed in order by the Peano

27 keys of their southwest corners; (3) for each parcel processed in this order, segment

28 (or node) entities are given entity identifiers in Peano key ordering. At higher layers,

29 segment and node records may be stored using a different ordering within the parcel,

30 such as by entity identifier. It is noted that if entity identifiers are used for ordering at

31 higher layers, the Peano key ordering may be inherited from the bottom layer. The

1   organization may be used with any of the parcelization processes disclosed above or

2   with any other parcelization process that organizes data spatially.

3   Figs. 7-10 illustrate application of the present embodiment to organize

4   geographic data within a parcel. Fig. 7 shows a map 300 of a portion of a geographic

5   region, such as the geographic region 12 of Fig. 1. Illustrated on the map are a

6   plurality of geographic features, such as road segments 22, end points 23 of road

7   segments, and positions 24 along road segments, which are represented in the map

8   database by road segment entities, node entities, and shape points, respectively. Also,

9   illustrated on the map 300 is a rectangle 302. The rectangle 302 represents an area

10   determined by a parcelization process, such as one of the parcelization processes

11   described above. The data entities (in a layer in a type of the geographic database)

12   which represent features encompassed within the rectangle 302 are grouped together

13   within a single parcel of the database. In the present embodiment, the parcel contains

14   routing layer 0 data, but in alternative embodiments other layers and types of data can

15   be used. Although only a few geographic features are shown in the area 300, it is

16   understood that in a typical map database formed by any of the processes mentioned

17   above, there may be hundreds or thousands of features or more represented in each

18   parcel. Also, although only one area 300 is shown which corresponds to one parcel, it

19   is understood that the present embodiment is applied to all areas that together make up

20   the entire geographic region represented by the map database, and further that the

21   present embodiment is applied in all parcels of at least one layer of at least one type.

22   According to the present embodiment, the data within the parcel are organized

23   to facilitate spatial searches. In the present embodiment, each of the rectangular

24   geographic areas (such as the area 302 shown in Fig. 7) that encompasses geographic

25   features which are represented by a separate collection of data entities contained in a

26   separate parcel is subdivided into a plurality of sub-areas 307. In one embodiment,

27   these sub-areas 307 are sub-rectangles and eight such sub-rectangles are formed from

28   the rectangular geographic area that encompasses the collection of data entities from

29   which each single separate parcel is formed. In the embodiment shown in Fig. 7, the

30   sub-rectangles 307 formed from the rectangle 302 are labeled 307(0) through 307(7).

31   (In alternative embodiments, more or fewer than eight sub-rectangles may be formed

1   although it is preferred that the number be equal to an integer power of 2, e.g. 4, 8, 16,

2   32, etc.)

3   In forming the sub-rectangles 307, boundaries of each sub-rectangle are

4   selected so that each of the sub-rectangles contains approximately the same number of

5   shape points and nodes as the other such sub-rectangles of the parcel. Shape point

6   information in the bottom layer is stored as a contiguous array of two- to five-byte

7   delta position records ranging from -32768 to +32767 navigation units (each of which

8   is equal to 1/100,000th of a degree, as mentioned above) from the position of the

9   previous shape point. Each segment record contains a count of shape point records

10   and a pointer to its first shape point.

11   In a present embodiment, the procedure used to separate the data within the

12   parcel may be similar to the procedure that was used to form the parcels. Any of the

13   procedures described above for determining the rectangular areas for forming parcels

14   may be used. In one process, cuts "C" are defined at any 1/32 division of a parcel

15   rectangle's minimum enclosing $2^i \times 2^j$ di-tile.   For example, a division exactly in half

16   is represented as a 16/32 cut. In Fig. 7, each cut, labeled C1, C2 . . . C7, represents a

17   division of the rectangular area 302 using this procedure to form the eight sub-

18   rectangles 307(0) through 307(7), each of which contains relatively the same amount

19   of data as the other sub-rectangles. Each sub-rectangle 307 is assigned an index 310

20   in the range 0-7.

21   This subdivision of the parcel's rectangular area 302 into sub-rectangles 307

22   may be represented by a kd-tree. (This kd-tree is used to represent the sub-rectangles

23   within a parcel and is separate from any kd-tree structure which may be used to

24   represent the division of data at the parcel level.) Fig. 8 graphically illustrates a

25   kd-tree 322 representing the sub-rectangles 307(0) through 307(7) of Fig. 7. The

26   kd-tree 322 of Fig. 8 also illustrates the division of the rectangle 302 into the sub-

27   rectangles 307 by the series of cuts C1 . . . C7.   Each of the leaf or internal nodes 326

28   of the kd-tree 322 corresponds to one of the sub-rectangles 307 so formed. (In this

29   embodiment, eight sub-rectangles 307 are formed and therefore the kd-tree 322 has

30   eight leaf nodes 326.) This kd-tree 322 is stored in the map database either in the

31   parcel itself or in an index file, such as in the index file 11 of Figs. 4 and 5, located

32   elsewhere in the database. Fig. 9 represents how the data in the kd-tree 322 are stored

1   in the map database. This kd-tree 322 may be used to narrow down the list of

2   segment records that need to be examined in a spatial search for segment records

3   within the parcel.

4   Referring to Fig. 10, the kd-tree 322 formed for each parcel is used to form a

5   data structure 320 for each parcel. The data structure 320 may be a bitmap, a bitmask,

6   or other type of index. Each segment record in the parcel is checked against each sub-

7   rectangle formed for its respective parcel. The segment record is assigned the index

8   310 of any sub-rectangle it intersects. A segment can intersect multiple sub-

9   rectangles, so an 8-bit bitmap (each bit of which corresponds to a sub-rectangle) is

10   associated with each segment. In Fig. 10, the data structure 320 is a bitmap. The

11   bitmap 320 includes entries for all the segments in the parcel formed from the data

12   entities encompassed within the rectangle 302 shown in Fig. 7. (For the sake of

13   clarity, only some of the segments are portrayed.) For example, segment 22(1) and

14   segment 22(2) are located entirely in sub-rectangle 307(0). These segments do not

15   intersect any sub-rectangles other than sub-rectangle 307(0). Accordingly, the entries

16   for these two segments in the bitmap 320 have bits only in the column associated with

17   the sub-rectangle 307(0). Likewise, segment 22(n) intersects only sub-rectangle

18   307(1) and accordingly, it is associated only with the sub-rectangle 307(1) with a bit

19   in the index 310 corresponding to the sub-rectangle 307(1). However, segment 22(g)

20   intersects both sub-rectangles 307(1) and 307(4), and accordingly it has bits

21   associated with both theses sub-rectangles in the bitmap 320. Segment 22(e)

22   intersects sub-rectangles 307(0), 307(2), 307(3), and 307(7). Accordingly, segment

23   22(e) has bits associated with these four sub-rectangles in the bitmap 320.

24   The data structure 320 is stored in the map database with the kd-tree 322 either

25   in the parcel itself or in an index file, such as in the index file 11 of Figs. 4 and 5,

26   located elsewhere in the database. The internal spatial index formed by the internal

27   kd-tree 322 and the bitmap 320 can be used to facilitate identification of data entity

28   records when searching within the parcel. If the geographic coordinates are provided,

29   the kd-tree 322 can be used to quickly identify which of the sub-rectangles 307

30   contains data encompassing the coordinates. Using the bitmap 320, the segment data

31   entities that intersect the sub-rectangle or sub-rectangles can quickly be identified.

32   Additional advantages follow when a search area encompasses more than one parcel.

1   Under these circumstances, using the bitmaps 320 associated with each of the

2   intersected parcels, the data entities in the sub-rectangles of each of the intersected

3   parcels can quickly be identified.

4

5   ***Example 1:***

6   An exemplary use of a geographic database incorporating the features of the

7   above embodiment is described in connection with Figs. 11 and 12. A navigation

8   system 500 uses a geographic database 530 stored on a medium 540 (such as a

9   CD-ROM, PCMCIA card, a DVD disk, or other suitable storage medium). The

10   navigation system 500 may be in a vehicle (not shown). The geographic database 530

11   is parcelized, i.e., it is separated into a plurality of parcels. Within each parcel 516 of

12   at least one layer of at least one type of geographic data, the data are arranged internal

13   of each of the parcels according to the embodiment described above and an internal

14   kd-tree index 322 and bitmap 320 are provided for each parcel so arranged.

15   According to this example, the end-user operating the navigation system 500

16   may want to have the area in which he/she is located displayed on a computer display

17   associated with the navigation system. Alternatively, the navigation system 500 may

18   require all the routing data in the vicinity of the navigation system 500 in order to

19   calculate a route to another location in the search area. In a still further alternative,

20   the end-user may have entered a destination location into the navigation system 500

21   and the navigation system 500 requires all the routing data records around the

22   destination location in order to calculate a route to the destination.

23   A navigation application program 520 is part of the navigation system 500.

24   The navigation application program 520 includes the software component portion of

25   the navigation system 500. The navigation application program 520 includes suitable

26   programming to define an appropriately-sized rectangular area based upon the

27   function being requested by the end-user. For example, if the end-user wants an

28   optimum route between a starting location and a destination location, the navigation

29   application program 520 may require all the routing data within a one kilometer

30   square area around both the starting location and the destination location. The

31   navigation application program 520 determines the coordinates of these suitably-sized

32   rectangular areas. The navigation application program 520 then accesses the medium

1 540 upon which the geographic database 530 is stored to obtain all the routing data

2 records that represent the road segments encompassed within the two rectangular

3 areas. In a present embodiment, the navigation application program 520 obtains the

4 data by calling an interface layer 550. The interface layer 550 is a library of software

5 programs that provides an interface between the navigation application software 520

6 and the geographic database 530. The interface layer 550 isolates the navigation

7 application program 520 from the complexity and evolution of the geographic

8 database 530. (The interface layer 550 is not essential to the present embodiment and

9 in an alternative implementation the navigation application program 520 may access

10 the geographic database 530 directly.)

11   Fig. 11 shows the same geographic area 300 as shown in Fig. 7. Some of the

12 geographic features area shown and the sub-rectangles 307(0) through 307(7) are

13 shown, but other features are omitted for the sake of clarity. Superimposed on the

14 area 300 is a search rectangle 400. The search rectangle 400 represents one of the

15. areas about which the navigation application program 520 requires data. For example,

16 the search rectangle may represent an area in which the vehicle is located. The search

17 rectangle 400 is defined by coordinates (x, y), (x+m, y), (x, y+n), and (x+m, y+n).

18 These coordinates are defined by the navigation application program 520 given a

19 vehicle position within the area of the rectangle 400.

20   Using an appropriate index, such as a global kd-tree index 556 that represents

21 the parcels 516, the parcel 516(n) that contains the data encompassed within the

22 geographic area 300 is identified, accessed, and read. This step may be performed by

23 a first routine 551 in the interface layer 550. Once the parcel 516(n) is read from the

24 media 540 upon which the geographic database 530 is stored, the data entities in the

25 parcel 516(n) may be in a compressed format. Thus, even though all the data in the

26 parcel 516(n) is in the memory of the navigation system 500, there are still

27 computational resources associated with de-compressing or otherwise processing the

28 data in the parcel to identify the data included in the parcel that represent the road

29 segment features contained within the boundaries of the search rectangle 400.

30 Although it is possible to de-compress and examine all the data in the parcel 516(n) to

31 identify the data that represents the road segment features within the search rectangle

32 400, this would involve de-compressing and examining many data records which do

1 not represent features in the search rectangle 400, resulting in wasted effort and

2 possibly slower performance. Using the parcel's internal kd-tree 322 and bitmap 320

3 that associates the parcel's data records with each of the sub-rectangles avoids this

4 wasted effort and may improve performance of the navigation system 500.

5   The global kd-tree search indicates which parcels are completely contained in

6 the search rectangle and which parcels partially intersect the search rectangle. The

7 internal kd-tree and spatial bitmaps need to be examined only for parcels which

8 partially intersect the search rectangle, because for parcels completely contained in the

9 search rectangle all data contained in the parcel *a fortiori* intersect the search

10 rectangle. Similarly, the search of a parcel's internal kd-tree indicates which of the

11 parcel's sub-rectangles are completely contained in the search rectangle and which of

12 the parcel's sub-rectangles partially intersect the search rectangle. Segments which

13 intersect sub-rectangles that are completely contained in the search rectangle *a fortiori*

14 intersect the search rectangle, so only those segments which intersect sub-rectangles

15 that partially intersect the search rectangle must be examined in further detail. This

16 detailed examination involves decompressing the segment record data and computing

17 the intersections of each successive segment portion between consecutive shape points

18 with the search rectangle. (This detailed examination of a segment can stop when the

19 intersection is non-empty thereby establishing that the segment intersects the search

20 rectangle).

21   Using the coordinates of the search rectangle 400 in the internal kd-tree 322

22 for the parcel 516(n), the sub-rectangles 307 within the parcel 516(n) that intersect the

23 search rectangle 400 can be identified. This step may be performed by a second

24 routine 552 in the interface layer 550. The second routine 552 compares the

25 geographic coordinates of the search rectangle 400 with the sub-rectangles 307(0)

26 through 307(7) to determine which sub-rectangles fall completely within the search

27 rectangle, which sub-rectangles fall completely outside the search rectangle, and

28 which sub-rectangles fall partially within the search rectangle. In the example shown

29 in Fig. 11, the search rectangle 400 is shown to intersect only the five sub-rectangles

30 307(2), 307(3), 307(5), 307(6), and 307(7). Using the bitmap 320, the segment data

31 records 22 within the parcel that intersect these sub-rectangles can be readily

32 identified. This step may be performed by a third routine 553 in the interface layer

550. In using the bitmap 320, the segment records that intersect these five

sub-rectangles can be identified by determining any segment entry that has at least

one bit on (e.g., "1") for the sub-rectangles in question. Only these segment records

are de-compressed and examined to determine whether the segments fall within the

search rectangle 400. The de-compression and examination can be performed by a

fourth routine 554 in the interface layer 550. The segment records that fall in the

other three sub-rectangles, 307(0), 307(1), and 307(4), do not have to be de-

compressed and examined, thereby avoiding unnecessary computation that may slow

down operation of the navigation system. Once the data records associated with the

five sub-rectangles that intersect the search rectangle are identified, they are de-

compressed and examined to determine whether they are encompassed within the

search rectangle 400. Not all the segment records in the five sub-rectangles are

necessarily encompassed within the search rectangle 400, so not all of them are

returned to the requesting routine in the navigation application 520. Use of the

internal kd-tree 322 and bitmap 320 considerably reduces the number of records that

have to be de-compressed and examined.

It is noted that using the internal kd-tree 322 and the bitmap 320 adds some

additional processing steps. However, the additional processing steps associated with

the use of the internal kd-tree and bitmap are generally more than offset by avoiding

the de-compression and examination of unnecessary segment records. It is noted that

use of the internal kd-tree 322 and bitmap 320 generally provides greatest advantages

when the search area 400 is relatively small compared to the rectangle 300 from

which the parcel is formed. This factor can be used to selectively skip using the

internal kd-tree 322 and bitmap 320 when the search rectangle is above a certain

predefined percentage of the size of the rectangle from which the parcel was formed.

Example 2:

Another exemplary use of a geographic database incorporating the features of

a present embodiment is described in connection with Figs. 12 and 13. As in the

previous example, the navigation system 500 uses the parcelized geographic database

530 and within each parcel 516 of at least one layer of at least one type of geographic

data, the data are arranged using sub-rectangular areas according to the embodiment

1   described above. As further mentioned above, an internal kd-tree index 322 and

2   bitmap 320 are provided for each parcel so arranged.

3   In this embodiment, the search rectangle requested by the navigation

4   application program 520 encompasses data entities in more than one parcel. Fig. 13

5   shows the same geographic area 300 as shown in Figs. 7 and 11, and in addition, Fig.

6   13 shows adjacent geographic areas 370, 380, and 390. Each of these geographic

7   areas is encompassed by a separate rectangle, 302, 372, 382, and 392, respectively.

8   The data encompassed by each of these rectangles are contained in separate parcels

9   516 in the geographic database 530. Although these rectangles 302, 372, 382, and

10   392 are shown to be the same size, they do not have to be.

11   Superimposed on the areas 300, 370, 380, and 390 is a search rectangle 402.

12   The search rectangle 402 represents the area about which the navigation system 500

13   requires data. The search rectangle 402 overlaps a part of each of the four areas 300,

14   370, 380, and 390.

15   Using the global kd-tree index 556 that represents all the parcels 516, the four

16   parcels that contain the data encompassed within the geographic areas 300, 370, 380,

17   and 390 can be accessed and read from the media 450 upon which the geographic

18   database 530 is stored. As noted in the previous example, the data entities in these

19   parcels may be in a compressed format. Thus, even though all the data in the four

20   parcels are in memory, there are still computational resources associated with de-

21   compressing or otherwise processing the data in these four parcels to identify which

22   of the data in these four parcels represent features contained within the search

23   rectangle 402. Using the internal kd-trees 322 and bitmaps 320 associated with each

24   of these parcels 516 avoids this wasted effort.

25   Using the coordinates of the search rectangle 402 with the internal kd-trees for

26   each of the parcels, the sub-rectangles within each rectangle represented by each of

27   the parcels that intersect the search rectangle 402 can be identified. This step may be

28   performed by the second routine 552 in the interface layer 550 which compares the

29   geographic coordinates of the search rectangle with the sub-rectangles of each of the

30   intersected parcels. As mentioned above, the first routine determines which sub-

31   rectangles fall completely within the search rectangle, which sub-rectangles fall

32   completely outside the search rectangle, and which sub-rectangles fall partially within

1   the search rectangle. In the example shown, the search rectangle 402 is shown to

2   intersect only two sub-rectangles 307(4) and 307(5) of the parcel represented by the

3   rectangle 302, three sub-rectangles 377(0), 377(1) and 377(2) of the parcel

4   represented by the rectangle 372, one sub-rectangle 387(7) of the parcel represented

5   by the rectangle 382, and one sub-rectangle 397(1) of the parcel represented by the

6   rectangle 392. It is noted that with respect to the parcel represented by the rectangle

7   372, one of the sub-rectangles 377(0) that intersects the search rectangle 402 falls

8   completely within the search rectangle. Then, using the bitmaps associated with each

9   parcel, the segment data records within the four parcels that intersect the sub-

10   rectangles that are partially within the search rectangle can be identified (i.e., 307(4),

11   307(5), 377(1), 377(2), 387(7), and 397(1)), the segment data records within the

12   parcel that intersects the sub-rectangle that is wholly within the search rectangle can

13   be readily identified (i.e., 377(0)), and the segment data records within the parcels that

14   do not intersect the sub-rectangles that are partially or wholly within the search

15   rectangle can be readily identified. This step may be performed by the third routine

16   553 in the interface layer 550. Only the segment records that are identified as

17   intersecting the sub-rectangles that are partially or wholly within the search rectangle

18   are de-compressed. Of these, the segment records that are identified as intersecting

19   the sub-rectangle which is wholly within the search rectangle (i.e. 377(0), are not

20   examined because these records are known to fall within the search rectangle and

21   therefore meet the search criterion. Only the segment records that are identified as

22   intersecting the sub-rectangles that are partially or wholly within the search rectangle

23   need to be both de-compressed and examined because some of these records may not

24   meet the search criterion of falling within the search rectangle 402. The segment

25   records in the sub-rectangles of each of these four parcels are de-compressed and

26   examined. The de-compression and examination can be performed by the fourth

27   routine 554 in the interface layer 550. The segment records that represent features

28   that intersect the search rectangle 402 are returned to the requesting routine in the

29   navigation application 520.

30   As demonstrated by this example, the arrangement of data within the parcels

31   and the indices associated with the arrangement enable a program to search the

32   database for records very efficiently. In the example, of the 32 sub-rectangles

1   included in the four parcels (i.e., four parcels with eight sub-rectangles per parcel),

2   only the segment records in six of the sub-rectangles had to both de-compressed and

3   examined to ascertain which of them fell within the search rectangle. The segment

4   records in one of the sub-rectangles had to be de-compressed, but did not have to be

5   examined since the sub-rectangle was determined to fall completely within the search

6   rectangle. The segment records in the remaining 25 sub-rectangles did not have to be

7   either de-compressed or examined. Avoiding de-compression or examination of these

8   many records can result in significant improvement in performance.

9

10   *Alternatives* Although the embodiments described above refer to routing data

11   segments, embodiments of the disclosed arrangement can be applied to other kinds of

12   data, such as cartographic data, point of interest data, or any other kind of data that is

13   spatially parcelized. Further, although the embodiments described above refer to

14   using the internal kd-tree indices and bitmaps with bottom layer data, these features

15   can be used with higher layers as well. In addition, although the embodiments

16   disclose the internal kd-tree and bitmap associated with each parcel as being located in

17   the parcel (e.g., physically close to the data in the parcel), the internal kd-tree and

18   bitmap for each parcel may be located physically outside the parcel in another location

19   in the database. For example, the internal kd-tree indices and bitmaps for each parcel

20   may be located with the global kd-tree or among the index files (11 of Figs. 4 and 5).

21   The advantages provided by the use of internal kd-tree indices and associated

22   bitmap may benefit certain kinds of applications. For example, in certain navigation

23   system platform implementations, the geographic database is located remotely from

24   the navigation application program and data from the geographic database is

25   transmitted via a communications link. This kind of navigation system would benefit

26   from an embodiment of the present system because fewer fully decompressed records

27   might need to be transmitted.

28   Another advantage provided by some of the disclosed embodiments is that the

29   parcel size may be made larger. In some systems, a larger parcel size may be desired.

30   If the parcel size used in a spatially organized geographic database were made larger,

31   larger geographic areas could be represented by the data records in each parcel.

32   Although this may provide performance enhancements for some kinds of navigating

1   functions, other functions may suffer performance decreases because more data are

2   being handled with each read access. The search rectangle function described in the

3   above examples, in which the data records that represent features in a search

4   rectangle, may be the kind of function whose performance is adversely affected by

5   larger parcel sizes. The embodiments disclosed above help reduce the effect of larger

6   parcels sizes for this kind of function, as well as other kinds of navigation functions.

7

8   It is intended that the foregoing detailed description be regarded as illustrative

9   rather than limiting and that it is understood that the following claims including all

10   equivalents are intended to define the scope of the invention.

11